

WHAT IS CLAIMED IS:

1                   1. An apparatus for executing a secure program in a computer system;  
2 wherein the ability to make workable copies of the secure program from the computer  
3 system is inhibited, the apparatus comprising:  
4                   a program memory in which the secure program data is stored in an  
5 encrypted form;  
6                   a security chip coupled to the program memory, the security chip  
7 comprising:  
8                   means for decrypting portions of the secure program into a clear  
9 portion and a remainder portion;  
10                  means for providing the clear portion to memory locations  
11 accessible by a processor; and  
12                  remainder memory for storing the remainder portion of the secure  
13 program, the remainder memory not directly accessible by the processor;  
14                  means for requesting subsets of the remainder portion for use by the  
15 processor; and  
16                  means, within the security chip, for checking that the requested subset is a  
17 subset expected to be requested given a stored state for the processor.

1                   2. The apparatus of claim 1, wherein the secure program stored in the  
2 program memory is stored with the program portion and the remainder portion stored  
3 separately.

1                   3. The apparatus of claim 1, wherein the remainder portion is a set of  
2 branch instructions of the secure program.

1                   4. The apparatus of claim 3, wherein the security chip further includes  
2 means for caching branch statements based on recently executed branches.

1                   5. The apparatus of claim 1, wherein the means for decrypting branches  
2 is configured with a decryption key.

1                   6. The apparatus of claim 5, wherein the decryption key is stored in  
2 volatile memory.

1                   7. The apparatus of claim 6, wherein the volatile memory is distributed  
2 over the security chip, the security chip further comprising overlying circuitry which  
3 overlies and obscures at least a part of the volatile memory.

1                   8. The apparatus of claim 7, wherein the overlying circuitry is coupled to  
2 a power source for the volatile memory such that the removal of the overlying circuitry  
3 removes the power to the overlying circuitry.

1                   9. The apparatus of claim 1, further comprising a clocking means, within  
2 the security chip, for determining a rate of instruction execution of the processor,  
3 wherein the security chip responds to processor requests only when the clocking means  
4 determines that the rate is within an expected range.

1                   10. The apparatus of claim 1, further comprising a data decompressor for  
2 decompressing the secure program after decryption, wherein the secure program is  
3 compressed before encryption.

1                   11. The apparatus of claim 10, wherein the decompressor is an entropy  
2 decoder.

1                   12. The apparatus of claim 1, further comprising a checksum means,  
2 within the security chip, for determining a checksum of bus accesses on a processor bus,  
3 wherein the security chip responds to processor requests only when the determined  
4 checksum matches an expected checksum.

1                   13. The apparatus of claim 1, further comprising a data scrambler for  
2 reordering data elements of the secure program according to a reversible and  
3 deterministic pattern determined by a key value, wherein the secure program is reordered  
4 by the inverse of the data scrambler before being placed in the program memory.

1           14. The apparatus of claim 13, wherein the data scrambler comprises a  
2 plurality of first-in, first-out buffers.

1           15. The apparatus of claim 13, wherein the reversible and deterministic  
2 pattern is generated by reference to the output of a pseudorandom number generator.

1           16. The apparatus of claim 1, wherein the means for decrypting operates  
2 based on the key value and the output of a pseudorandom number generator.

1           17. The apparatus of claim 1, further comprising means for altering the  
2 operation of the security chip and the flow of the program when said means for checking  
3 detects that an unexpected subset has been requested, where by the altered operation  
4 causes a negative effect on the program flow or operation.

1           18. The apparatus of claim 17, wherein the means for altering is a  
2 means for halting the processor.

1           19. An apparatus for securing program data from unauthorized  
2 copying, comprising;  
3           a branch separator for extracting branch statements from the program data;  
4           a compressor for compressing the extracted branch statements and a  
5 remainder of the program data to form compressed data; and  
6           an encryptor for encrypting the compressed data.

1           20. The apparatus of claim 19, wherein the branch separator comprises:  
2           means for automatically generating checksum data representing checksums  
3 of program data; and  
4           means for automatically generating timing information used to assess  
5 timing of program data processing,  
6           whereby the checksum data and the timing information are compressed by  
7 the compressor and encrypted by the encryptor.

1           21. A method of executing a secure program to prevent copying of the  
2 secure program in a usable form from information acquired over an insecure processor  
3 bus, comprising the steps of:

4           accepting a request from the insecure processor for a block of program  
5 data, the block of program data including at least one of one or more program  
6 instructions or one or more program data elements;

7           decrypting, in a secure manner, the block of program data into a clear  
8 portion and a remainder portion;

9           providing the clear portion to the insecure processor; and

10          accepting requests from the insecure processor for elements of the  
11 remainder portion;

12          checking that the request is proper given the state of the insecure processor  
13 and previous requests;

14          processing the requests from the insecure processor for elements of the  
15 remainder portion; and

16          responding to the requests, wherein underlying remainder portion elements  
17 are not feasibly determined by reference to only the information content of a response to  
18 a request.

1           22. The method of claim 21, further comprising the steps of:

2           separating a program into the clear portion and the remainder portion to  
3 form a secure program; and

4           encrypting the secure program prior to placing the secure program in a  
5 memory accessible by attackers intent on making unauthorized copies of the secure  
6 program.

1           23. The method of claim 22, wherein the step of separating is a step of

2 separating branch instructions of the secure program from other instructions of the secure  
3 program.

1           24. The method of claim 21, wherein the step of decrypting is performed

2 with a decryption key.

1                   25. The method of claim 24, further comprising the step of storing the  
2 decryption key in volatile memory.

1                   26. The method of claim 25, further comprising the steps of:  
2 providing a power source to the volatile memory;  
3 covering the volatile memory with a circuit such that the power source is  
4 removed from the volatile memory when the circuit is disturbed and the contents of the  
5 volatile memory cannot be easily measured without removing the circuit.

1                   27. The method of claim 21, further comprising the step of checking a  
2 rate of instruction execution of the processor prior to providing a response to a request  
3 for information.

1                   28. The method of claim 21, further comprising the step of  
2 decompressing the secure program after decryption, wherein the secure program is  
3 compressed before encryption.

1                   29. The method of claim 21, further comprising the steps of:  
2 determining a checksum of bus accesses on a processor bus;  
3 comparing the checksum to a precalculated checksum expected for the  
4 instructions of the secure program which were expected to be executed; and  
5 preventing the unobstructed operation of the secure program when the  
6 checksum and the precalculated checksum differ.

1                   30. The method of claim 21, further comprising a steps of:  
2 scrambling an order of data elements of the secure program according to a  
3 reversible and deterministic pattern determined by a key value prior to storage in a  
4 memory accessible by attackers; and  
5 descrambling the order of the data elements upon proper request of the  
6 processor.

1                   31. The method of claim 30, wherein the step of scrambling comprises a  
2 step of generating a pseudorandom number used to form the reversible and deterministic  
3 pattern.

1                   ~~32.~~ A method for securing a program against unauthorized copying,  
2 comprising the steps of:  
3                   separating program code according to sequences of nonbranch instructions  
4 and branch instructions;  
5                   compressing the non-branch instructions to form a first set of compressed  
6 data;  
7                   compressing the branch instructions to form a second set of compressed  
8 data; and  
9                   encrypting the first and second sets of compressed data.